CLAIMS

What is claimed is:

Sub B1

1. A method of processing a read request comprising the acts of:

transmitting a first read request from a requesting device, the first read request comprising a first system address;

receiving the first read request at a memory controller;

mapping the first system address into a first memory address, the first memory address comprising a first chip select, a first bank address, a first row address and a first column address; and

sorting the first read request by one of the first chip select and the first bank address such that the first read request is injected into a first read queue.

2. The method of processing a read request, as set forth in claim 1, comprising the acts of:

receiving a second read request at the memory controller, the second read request having a second system address and being received immediately subsequent to the first read request;

mapping the second system address into a second memory address, the second memory address comprising a second chip select, a second bank address, a second row address and a second column address; and

sorting the second read request by one of the corresponding second chip select and second bank address such that the second read request is injected into a second read queue if the one of the first chip select and first bank address corresponding to the first read request is different as the one of the second chip select and second bank address corresponding to the second read request and wherein the second read queue is different from the first read queue.

3. The method of processing a read request, as set forth in claim 2, comprising the act of alternately selecting read requests from the first read queue and the second read queue such that back-to-back requests are alternately processed between the first read queue and the second read queue.

4. The method of processing a read request, as set forth in claim 3, wherein the act of alternately selecting read requests is performed by an arbiter.

5. The method of processing a read request, as set forth in claim 2, comprising the act of selecting read requests to be processed such that back-to-back requests are not processed to the same chip select.

6. The method of processing a read request, as set forth in claim 2, comprising the act of selecting read requests to be processed such that back-to-back requests are not processed to the same bank address.

7. The method of processing a read request, as set forth in claim 2, wherein the second chip select corresponding to the second read request identifies a corresponding dual inline memory module (DIMM) to which the second read request is directed.

8. The method of processing a read request, as set forth in claim 2, wherein the second bank address corresponding to the second read request identifies a corresponding bank in a corresponding dual inline memory module (DIMM) to which the second read request is directed.

9. The method of processing a read request, as set forth in claim 1, wherein the first chip select corresponding to the first read request identifies a corresponding dual inline memory module (DIMM) to which the first read request is directed.

10. The method of processing a read request, as set forth in claim 1, wherein the first bank address corresponding to the first read request identifies a corresponding bank in a corresponding dual inline memory module (DIMM) to which the first read request is directed.

11. A method of processing read requests in a redundant memory system comprising the acts of:

(a) receiving a plurality of read requests at a memory controller, each read request having a corresponding chip select and bank address ;

(b) inserting the plurality of read requests into one or more queues; and

(c) prioritizing the processing of the plurality of read requests such that back-to-back read requests are not directed to the same one of a corresponding chip select and a bank address.

12. The method of processing read requests in a redundant memory system, as set forth in claim 11, wherein act (b) comprises the act of inserting the plurality of read requests into one of a first queue and a second queue.

13. The method of processing read requests in a redundant memory system, as set forth in claim 12, wherein act (c) comprises the act of alternately selecting read requests from the first queue and the second queue such that back-to-back requests are alternately processed between the first queue and the second queue.

14. The method of processing read requests in a redundant memory system, as set forth in claim 13, wherein the act of alternately selecting read requests is performed by an arbiter.

15. The method of processing read requests in a redundant memory system, as set forth in claim 11, wherein the chip select identifies a corresponding dual inline memory module (DIMM) to which a corresponding one of the plurality of read requests is directed.

16. The method of processing read requests in a redundant memory system, as set forth in claim 11, wherein the bank address identifies a corresponding bank in a corresponding

dual inline memory module (DIMM) to which a corresponding one of the plurality of read requests is directed.

17. A memory cartridge comprising:

a plurality of memory devices;

a memory controller operably coupled to the plurality of memory devices and configured to receive a plurality of read requests from a requesting device, each of the plurality of read requests comprising a chip select and a bank address, wherein the memory controller is further configured to sort the plurality of read requests by one of the corresponding chip select and bank address;

a first and second read queues each operably coupled to the memory controller and each configured to store the read requests sorted by the memory controller; and

an arbiter operably coupled to each of the first and second read queues and configured to select read requests stored in the queues such that consecutive read requests do not have the same one of the chip select and the bank address.

18. The memory cartridge, as set forth in claim 17, wherein the plurality of memory devices comprises a plurality of synchronous dynamic random access memory (SDRAM) devices.

19. The memory cartridge, as set forth in claim 17, wherein the arbiter is configured to select consecutive read requests alternately between the first and second read queues.

20. A method of processing requests comprising the acts of:

(a) storing a first request in a first storage device during a first clock cycle;

(b) transmitting the first request from the first storage device during a second clock cycle, the second clock cycle being immediately subsequent to the first clock cycle;

(c) storing a second request in a second storage device during the second clock cycle;

(d) transmitting the second request from the second storage device during a third clock cycle, the third clock cycle being immediately subsequent to the second clock cycle; and

(e) storing a third request in the first storage device during the third clock cycle.

21. The method of processing requests, as set forth in claim 20, wherein act (a) comprises the act of storing the first request in a first latch during the first clock cycle.

22. The method of processing requests, as set forth in claim 20, wherein act (c) comprises the act of storing the second request in a second latch during the second clock cycle.

23. The method of processing requests, as set forth in claim 20, wherein act (b) comprises the acts of:

transmitting the first request to a memory controller; and

processing the first request by the memory controller.

24. The method of processing requests, as set forth in claim 23, wherein act (d) comprises the acts of:

transmitting the second request to a memory controller; and

processing the second request immediately subsequent to the processing of the first request.

25. A method of sorting requests comprising the acts of:

(a) arbitrating between a first read request and a first write request to produce a first output, one of the first read request and the first write request being a higher priority request and the other being a lower priority request, the first output corresponding to the higher priority request;

(b) storing the first output in a first storage device;

(c) arbitrating between the lower priority request and one of a second read request and a second write request to produce a second output;

(d) storing the second output in a second storage device; and

(e) processing the first output from the first device and the second output from the second storage device on immediately subsequent clock cycles.

26. The method of sorting requests, as set forth in claim 25, comprising the acts of:

receiving a plurality of read requests from a read command queue; and arbitrating among the plurality of read requests to produce the first read request, the first read request corresponding to a highest priority read request received from the read command queue.

27. The method of sorting requests, as set forth in claim 25, comprising the acts of:

receiving a plurality of write requests from a write command queue; and

arbitrating among the plurality of write requests to produce the first write request, the first write request corresponding to a highest priority write request received from the write command queue.

28. The method of sorting requests, as set forth in claim 25, wherein act (b) comprises the act of storing the first output in a first latch.

29. The method of sorting requests, as set forth in claim 25, wherein act (d) comprises the act of storing the second output in a second latch.

30. A method of sorting requests in an arbiter comprising the acts of:

(a) arbitrating among a plurality of read requests in one or more read queues to produce a first priority read request;

(b) arbitrating among a plurality of write requests in one or more write queues to produce a first priority write request;

(c) arbitrating between the priority read request and the priority write request to produce a first priority request the first priority request corresponding to the higher priority of the priority read request and the priority write request;

(d) storing the first priority request in a first storage device during a first clock cycle;

(e) arbitrating between the lower priority of the first priority read request and the first priority write request and one of a second priority read request and a second priority write request to produce a second priority request;

(f) storing the second priority request in a second storage device during a second clock cycle, the second clock cycle being immediately subsequent to the first clock cycle;

(g) transmitting the first priority request from the first storage device to a memory controller; and

(h) transmitting the second priority request from the second storage device to the memory controller.

31. The method of sorting requests, as set forth in claim 30, wherein acts (b) and (c) occur simultaneously.

32. The method of sorting requests, as set forth in claim 30, wherein acts (f) and (g) occur simultaneously.

33. The method of sorting requests, as set forth in claim 30, wherein acts (g) and (h) occur on immediately successive clock cycles.

34. A memory cartridge comprising:

write arbitration select logic configured to determine an order of priority of a plurality of write requests and to produce a first ordered output stream based on the priority determination;

read arbitration select logic configured to determine an order of priority of a plurality of read requests and to produce a second ordered output stream based on the priority determination;

request arbitration select logic operably coupled to each of the read arbitration select logic and the write arbitration select logic and configured to receive each of the first ordered output and the second ordered output and to produce a priority output stream based on a priority determination between the first ordered output stream and the second ordered output stream;

selection logic coupled to the output of the request arbitration select logic and configured alternately to direct the priority output stream to one of a first and second selection output; and

a first and a second storage device, each storage device coupled to the selection logic and configured alternately to receive the priority output stream from the selection logic.

35. The memory cartridge, as set forth in claim 34, comprising a write command queue configured to provide the plurality of write requests to the write arbitration select logic.

36. The memory cartridge, as set forth in claim 34, comprising a read command queue configured to provide the plurality of read requests to the read arbitration logic.

37. The memory cartridge, as set forth in claim 34, comprising a memory controller operably coupled to an output of each of the first and second storage device and configured to process the outputs from the first and second storage device.

38. The memory cartridge, as set forth in claim 34, wherein each of the first and second storage devices comprises a latch.

39. A method of storing data comprising the acts of:

(a) transmitting a cacheline of data from a device;

(b) parsing the cacheline of data into a plurality of data segments;

(c) delivering each of the plurality of data segments to a respective memory controller; and

(d) storing each of the plurality of data segments in a corresponding memory device.

40. The method of storing data, as set forth in claim 39, wherein act (a) comprises the act of transmitting a 32-byte cacheline of data.

41. The method of storing data, as set forth in claim 39, wherein act (b) comprises the act of parsing the cacheline of data into four data segments.

42. The method of storing data, as set forth in claim 39, wherein act (c) comprises the act of delivering each of the plurality of data segments via a different memory bus with respect to each other.

43. The method of storing data, as set forth in claim 39, wherein act (c) comprises the act of delivering each of the data segments to a respective memory controller simultaneously.

44. The method of storing data, as set forth in claim 39, wherein the act of storing each of the plurality of data segments occurs in parallel.

45. A system comprising:

a host controller configured to stripe a cacheline of data into a plurality of data segments;

a plurality of memory cartridges coupled to the host controller, each cartridge comprising a memory controller, wherein each of the memory controllers is configured to receive a single data segment from the cacheline of data; and

a plurality of memory devices coupled to a corresponding memory controller and configured to store the data segment transmitted from the respective memory controller.

46. The system, as set forth in claim 45, comprising five memory cartridges.

47. The system, as set forth in claim 45, wherein the memory cartridges comprise a redundant array of memory modules.

48. The system, as set forth in claim 45, wherein each of the memory controllers is configured to receive the single data segment simultaneously with respect to each other.

49. The system, as set forth in claim 45, wherein each of the plurality of memory devices comprise a synchronous dynamic random access memory (SDRAM) device.

50. A computer system comprising:

a processor; and

a memory system coupled to the processor and comprising a plurality of configuration registers, wherein each configuration register is configured to receive one or more bits corresponding to a system address and further configured to convert the one or more bits into one of a memory bank address, a memory row address, and a memory column address.

51. The computer system, as set forth in claim 50, wherein the memory system comprises a redundant array of memory modules.

52. The computer system, as set forth in claim 50, wherein the memory system comprises a plurality of memory controllers, and wherein each of the plurality of memory controllers comprises a configuration register corresponding to the bank address, a configuration

register corresponding to the row address, and a configuration register corresponding to the column address.

53. The computer system, as set forth in claim 50, wherein each of the plurality of configuration registers is programmable.

54. A memory system comprising a plurality of memory controllers, wherein each of the plurality of memory controllers comprises:

a first configuration register corresponding to a memory bank address;

a second configuration register corresponding to a memory row address; and

a third configuration register corresponding to a memory column address.

55. The memory system, as set forth in claim 54, wherein the first configuration register comprises an upper boundary register and a lower boundary register.

56. The memory system, as set forth in claim 54, wherein the second configuration register comprises an upper boundary register and a lower boundary register.

57. The memory system, as set forth in claim 54, wherein the third configuration register comprises an upper boundary register and a lower boundary register.

58. The memory system, as set forth in claim 54, wherein the first configuration register comprises a programmable register.

59. The memory system, as set forth in claim 54, wherein the second configuration register comprises a programmable register.

60. The memory system, as set forth in claim 54, wherein the third configuration register comprises a programmable register.

61. A memory controller comprising:

a first configuration register configured to provide a bank address corresponding to one of a memory read request and a memory write request;

a second configuration register configured to provide a row address corresponding to the one of the memory read request and the memory write request; and

a third configuration register configured to provide a column address corresponding to the one of the memory read address and the memory write address.

62. The memory controller, as set forth in claim 61, wherein the first configuration register comprises a programmable register.

63. The memory controller, as set forth in claim 61, wherein the second configuration register comprises a programmable register.

64. The memory controller, as set forth in claim 61, wherein the third configuration register comprises a programmable register.

65. The memory controller, as set forth in claim 61, wherein the first configuration register comprises an upper boundary register and a lower boundary register.

66. The memory controller, as set forth in claim 61, wherein the second configuration register comprises an upper boundary register and a lower boundary register.

67. The memory controller, as set forth in claim 61, wherein the third configuration register comprises an upper boundary register and a lower boundary register.

68. A system comprising:

a host/data controller configured to deliver requests, wherein each of the requests comprises a system address; and

a memory controller operably coupled to the host/data controller and configured to receive the requests from the host/data controller, wherein the host/data controller comprises a programmable mapping register, wherein the programmable mapping register is configured to convert the system address of each request into a memory address.

69. The system, as set forth in claim 68, wherein the memory address comprises each of a bank address, a row address, and a column address.

70. The system, as set forth in claim 69, wherein the programmable mapping register is configured to define the bit placement of each of the bank address, the row address, and the column address.